

**METHODS & APPARATUS FOR SYNCHRONIZING &  
PROPAGATING DISTRIBUTED ROUTING DATABASES**

*BY INVENTORS Puneet Agarwal, Bora Akyol, Erol Basturk, Mike Mussoline,  
& Russ Tuck*

5

**BACKGROUND OF THE INVENTION**

10    1.    **Field of the Invention**

          The present invention relates to methods and apparatus for synchronizing  
and propagating distributed routing databases. The invention also relates to  
methods for distributing routing data within a distributed processor router  
15    system.

          2.    **Background of the Related Art**

          In the context of internetworking, routing is the coordinated transfer  
20    of information from a source to a destination via hardware known as a  
router. Routing occurs at Layer 3, the network layer, of the OSI reference  
model of the ISO (International Society for Standardization). The OSI  
reference model is a conceptual model composed of seven layers, each  
specifying particular network functions. The two lowest layers (layers 1 and  
25    2) of the OSI model, namely the physical and data link layers, are  
implemented in both hardware and software. Layer 3 and layers upwards  
therefrom are generally implemented only in software.

Using terminology of the International Organization for Standardization (ISO), network devices may be classified as follows. Those devices with the capability to forward packets between subnetworks are referred to as *intermediate systems* (ISs). (In contrast, network devices without such capabilities are called *end systems*). Intermediate systems may be classified as *intradomain* ISs, i.e., those which can communicate within routing domains, and *interdomain* ISs which can communicate both within and between routing domains. A *routing domain*, or *autonomous system*, can be considered to be a part of an internetwork which is regulated under common administrative authority.

A key component of routing is determination of optimal routing paths for data packets. Thereafter a second component, which may be referred to as "forwarding", comprises transporting packets through the internetwork. Determination of optimal routing paths relies on one or more routing protocols to provide and update a routing database for each router in a network. Depending on the particular routing protocol(s) used, various metrics are involved in building the routing database. Metrics that may be used by various routing protocols, either singly or as components of hybrid metrics, include: bandwidth, cost, path length, reliability, and load. Such metrics are well known in the art.

Routing protocols are used to determine best routes for transporting packets through an internetwork. Routing in a network can be classified as either dynamic or static. Static routing is accomplished by using table mappings which are entered by a user (e.g. network administrator) prior to routing, and are only changed by user input. Dynamic routing is accomplished by routing protocols that adjust to changing network

conditions in response to incoming route update information. As a result, routes are recalculated, new routing update messages are sent out to peer routers, and updated routing databases are constructed. Routing protocols may be interior or exterior. Conventionally, interior routing protocols are used for determining routes within a routing domain. Examples of interior routing protocols are Routing Information Protocol (RIP) and Open Shortest Path First (OSPF). Exterior routing protocols exchange routing information between routing domains. Examples of exterior routing protocols are Border Gateway Protocol (BGP) and Exterior Gateway Protocol (EGP).

OSPF is a unicast routing protocol that requires each router in a network to be aware of all available links in the network. OSPF calculates routes from each router running the protocol to all possible destinations in the network. Intermediate System to Intermediate System (IS-IS) is an OSI link-state hierarchical routing protocol based on DECnet Phase V routing, whereby ISs (routers) exchange routing information based on a single metric, to determine network topology.

BGP performs interdomain routing in TCP/IP networks. As an exterior gateway protocol (EGP), BGP performs routing between multiple routing domains and exchanges routing and reachability information with other BGP systems. Each BGP router maintains a routing database that lists all feasible paths to a particular network. The router does not refresh the routing database, however. Instead, routing information received from peer routers is retained until an incremental update is received. BGP devices exchange routing information upon initial data exchange and after

incremental updates. When a router first connects to the network, BGP routers exchange their entire BGP routing tables.

5 In order to update their routing databases, routers send and receive information regarding network topology. Examples of such information include routing update messages, and link-state advertisements. By communicating with other routers in this way, each router obtains a routing database that defines the current topology of the network of which it is a part, enabling determination of optimal routing path.

10

Entries are added to and removed from the route database either by the user (e.g., a network administrator) in the form of static routes, or by various dynamic routing protocol tasks. In dynamic routing, routes are updated by software running in the router. The routing database defines a mapping from destination address to logical (output) interface, enabling the router to forward packets along the best route toward their destination. The route database is also the principal medium used to share routes among multiple active routing protocols. Thus, the routing database comprises an essential entity at the heart of every router.

20

Typically, two or three routing protocols may be active in any one router. The routing database as such is a superset of the set of routes actually used for forwarding packets. This is due, in part, to the fact that different routing protocols compute their preferred routes independently of each other, based on different metrics. Only when all route entries generated by the full complement of routing protocols are shared in the routing database, or route table, can the best routes be selected. The result of this selection is a subset of the routing database commonly referred to as the

forwarding table. The forwarding table can be considered a filtered view of the routing database. The forwarding table is used by all entities of the router that have to forward packets in and out of the router.

5           In conventional or prior art non-scalable routers, which have a modest number of interfaces, there is a single copy of the routing database shared by all of the routing protocols. In non-scalable routers, the computational power available to the routing protocols is typically limited to a single processor. Also, in non-scalable routers, the number of entities  
10       requiring a copy of the forwarding table is relatively small.

          In contrast, in routers with a relatively large number of interfaces, a possibility exists for imposing much higher computational loads on the processor, up to a point where it is no longer feasible to run all routing  
15       protocols on the same processor. In order to realize improved performance from such routers, the protocol computational load must be distributed onto a plurality of processors. Furthermore, in routers with a very large number of interfaces, the number of entities requiring a copy of the forwarding table can be very large, for example, numbering several thousands. This latter  
20       situation also imposes higher computational loads and the need for a plurality of processors per router.

          However, running the routing protocols on a plurality of processors, each processor having a copy of the routing database, introduces a potential  
25       problem into the routing system. The problem is the critical requirement to keep all copies of the routing database consistent. This requirement is critical because the view of the routing database presented to the routing protocols is vital to correct routing. Moreover, the ability to provide an accurate and

timely copy of the forwarding table to a very large number of entities in the system is necessary in order to leverage the benefits provided by a distributed routing database environment.

- 5           The instant invention provides a method for the distribution and synchronization of the routing database and forwarding table to a large number of entities within a distributed processor environment of a scalable router.

10

## **SUMMARY OF THE INVENTION**

- According to one aspect of the invention, there is provided a method for the synchronized distribution of routing data in a distributed processor  
15 router. The invention allows multiple routing databases, formed by distributed routing protocols, to be synchronized. The invention further allows distributed propagation of the synchronized database.

- One feature of the invention is that it enables the scaling of routing protocol tasks instantiated on multiple processors. Another feature of the  
20 invention is that it provides a distributed processor router environment, in which a plurality of processors host at least one of a plurality of different routing protocols. Another feature of the invention is that it provides a route table manager for controlling the propagation of a synchronized routing database within a distributed processor environment.

25

          One advantage of the invention is that it allows routing databases to be constructed and propagated in a distributed manner by instantiating routing protocol tasks on multiple processors. Another advantage of the

invention is that it provides a method for exchanging route data between a plurality of processors within a distributed processor router environment, wherein the exchange of route data is controlled by a route table manager (RTM). Another advantage of the invention is that it provides a method for

5 registering a first RTM task as a client of a second RTM task in order to establish a first RTM task-second RTM task client-server relationship, wherein the first RTM task and the second RTM task occupy either the same hierarchical level or different hierarchical levels. Another advantage of the invention is that it provides a method for establishing a first RTM task-

10 second RTM task client-server relationship, wherein the first RTM task is running on a line card of a highly scalable router, and the second RTM task is running on a control card of the same router.

These and other advantages and features are accomplished by the

15 provision of a method of synchronized distribution of routing data in a distributed processor router, including the following steps: a) running zero or more routing protocols of a complement of routing protocols on each of a first plurality of processors, wherein each routing protocol of the complement of routing protocols generates routing data; b) registering each

20 of the first plurality of processors with at least one other of the first plurality of processors; c) exchanging the routing data between members of the first plurality of processors, such that each of the first plurality of processors receives a full complement of routing data generated by the complement of routing protocols, the complement of routing data providing a complete

25 routing database; d) forming a forwarding database from the complete routing database, the forwarding database comprising a subset of the complete routing database; and e) propagating the forwarding database from



1. **Introduction**  
 2. **Background**  
 3. **Methodology**  
 4. **Results**  
 5. **Discussion**  
 6. **Conclusion**  
 7. **References**  
 8. **Appendix**  
 9. **Index**  
 10. **Table of Contents**  
 11. **Abstract**  
 12. **Summary**  
 13. **Key Words**  
 14. **Keywords**  
 15. **Subject Headings**  
 16. **Classification**  
 17. **Indexing**  
 18. **References**  
 19. **Appendix**  
 20. **Index**  
 21. **Table of Contents**  
 22. **Abstract**  
 23. **Summary**  
 24. **Key Words**  
 25. **Keywords**  
 26. **Subject Headings**  
 27. **Classification**  
 28. **Indexing**  
 29. **References**  
 30. **Appendix**  
 31. **Index**  
 32. **Table of Contents**  
 33. **Abstract**  
 34. **Summary**  
 35. **Key Words**  
 36. **Keywords**  
 37. **Subject Headings**  
 38. **Classification**  
 39. **Indexing**  
 40. **References**  
 41. **Appendix**  
 42. **Index**  
 43. **Table of Contents**  
 44. **Abstract**  
 45. **Summary**  
 46. **Key Words**  
 47. **Keywords**  
 48. **Subject Headings**  
 49. **Classification**  
 50. **Indexing**  
 51. **References**  
 52. **Appendix**  
 53. **Index**  
 54. **Table of Contents**  
 55. **Abstract**  
 56. **Summary**  
 57. **Key Words**  
 58. **Keywords**  
 59. **Subject Headings**  
 60. **Classification**  
 61. **Indexing**  
 62. **References**  
 63. **Appendix**  
 64. **Index**  
 65. **Table of Contents**  
 66. **Abstract**  
 67. **Summary**  
 68. **Key Words**  
 69. **Keywords**  
 70. **Subject Headings**  
 71. **Classification**  
 72. **Indexing**  
 73. **References**  
 74. **Appendix**  
 75. **Index**  
 76. **Table of Contents**  
 77. **Abstract**  
 78. **Summary**  
 79. **Key Words**  
 80. **Keywords**  
 81. **Subject Headings**  
 82. **Classification**  
 83. **Indexing**  
 84. **References**  
 85. **Appendix**  
 86. **Index**  
 87. **Table of Contents**  
 88. **Abstract**  
 89. **Summary**  
 90. **Key Words**  
 91. **Keywords**  
 92. **Subject Headings**  
 93. **Classification**  
 94. **Indexing**  
 95. **References**  
 96. **Appendix**  
 97. **Index**  
 98. **Table of Contents**  
 99. **Abstract**  
 100. **Summary**  
 101. **Key Words**  
 102. **Keywords**  
 103. **Subject Headings**  
 104. **Classification**  
 105. **Indexing**  
 106. **References**  
 107. **Appendix**  
 108. **Index**  
 109. **Table of Contents**  
 110. **Abstract**  
 111. **Summary**  
 112. **Key Words**  
 113. **Keywords**  
 114. **Subject Headings**  
 115. **Classification**  
 116. **Indexing**  
 117. **References**  
 118. **Appendix**  
 119. **Index**  
 120. **Table of Contents**  
 121. **Abstract**  
 122. **Summary**  
 123. **Key Words**  
 124. **Keywords**  
 125. **Subject Headings**  
 126. **Classification**  
 127. **Indexing**  
 128. **References**  
 129. **Appendix**  
 130. **Index**  
 131. **Table of Contents**  
 132. **Abstract**  
 133. **Summary**  
 134. **Key Words**  
 135. **Keywords**  
 136. **Subject Headings**  
 137. **Classification**  
 138. **Indexing**  
 139. **References**  
 140. **Appendix**  
 141. **Index**  
 142. **Table of Contents**  
 143. **Abstract**  
 144. **Summary**  
 145. **Key Words**  
 146. **Keywords**  
 147. **Subject Headings**  
 148. **Classification**  
 149. **Indexing**  
 150. **References**  
 151. **Appendix**  
 152. **Index**  
 153. **Table of Contents**  
 154. **Abstract**  
 155. **Summary**  
 156. **Key Words**  
 157. **Keywords**  
 158. **Subject Headings**  
 159. **Classification**  
 160. **Indexing**  
 161. **References**  
 162. **Appendix**  
 163. **Index**  
 164. **Table of Contents**  
 165. **Abstract**  
 166. **Summary**  
 167. **Key Words**  
 168. **Keywords**  
 169. **Subject Headings**  
 170. **Classification**  
 171. **Indexing**  
 172. **References**  
 173. **Appendix**  
 174. **Index**  
 175. **Table of Contents**  
 176. **Abstract**  
 177. **Summary**  
 178. **Key Words**  
 179. **Keywords**  
 180. **Subject Headings**  
 181. **Classification**  
 182. **Indexing**  
 183. **References**  
 184. **Appendix**  
 185. **Index**  
 186. **Table of Contents**  
 187. **Abstract**  
 188. **Summary**  
 189. **Key Words**  
 190. **Keywords**  
 191. **Subject Headings**  
 192. **Classification**  
 193. **Indexing**  
 194. **References**  
 195. **Appendix**  
 196. **Index**  
 197. **Table of Contents**  
 198. **Abstract**  
 199. **Summary**  
 200. **Key Words**  
 201. **Keywords**  
 202. **Subject Headings**  
 203. **Classification**  
 204. **Indexing**  
 205. **References**  
 206. **Appendix**  
 207. **Index**  
 208. **Table of Contents**  
 209. **Abstract**  
 210. **Summary**  
 211. **Key Words**  
 212. **Keywords**  
 213. **Subject Headings**  
 214. **Classification**  
 215. **Indexing**  
 216. **References**  
 217. **Appendix**  
 218. **Index**  
 219. **Table of Contents**  
 220. **Abstract**  
 221. **Summary**  
 222. **Key Words**  
 223. **Keywords**  
 224. **Subject Headings**  
 225. **Classification**  
 226. **Indexing**  
 227. **References**  
 228. **Appendix**  
 229. **Index**  
 230. **Table of Contents**  
 231. **Abstract**  
 232. **Summary**  
 233. **Key Words**  
 234. **Keywords**  
 235. **Subject Headings**  
 236. **Classification**  
 237. **Indexing**  
 238. **References**  
 239. **Appendix**  
 240. **Index**  
 241. **Table of Contents**  
 242. **Abstract**  
 243. **Summary**  
 244. **Key Words**  
 245. **Keywords**  
 246. **Subject Headings**  
 247. **Classification**  
 248. **Indexing**  
 249. **References**  
 250. **Appendix**  
 251. **Index**  
 252. **Table of Contents**  
 253. **Abstract</**

15           These and other advantages and features of the invention will be set forth in part in the description which follows and in part will become apparent to those having ordinary skill in the art upon examination of the following, or may be learned from practice of the invention. The advantages of the invention may be realized and attained as particularly pointed out in the appended claims.

## BRIEF DESCRIPTION OF THE DRAWINGS

Figs. 1A, 1B and 1C are block diagrams showing basic architecture of a scalable router according to an embodiment of the invention;

5        Fig. 2 schematically represents exchange of route data generated by different routing protocols, according to an embodiment of the invention.

      Fig. 3 schematically represents exchange of route data generated by two different routing protocols showing four servers and two clients, according to an embodiment of the invention.

10       Fig. 4 schematically represents chronology of RTM-mediated data flow between two control cards, according to one embodiment of the invention.;

      Fig. 5 schematically represents a hierarchical relationship of RTM tasks according to a preferred embodiment of the invention.

15       Fig. 6 schematically represents a hierarchical relationship between route table manager tasks, according to one embodiment of the invention;

      Fig. 7A schematically represents the distribution of route data from a route table manager Level-1 task primary server to a route table manager Level-2 task client, according to the invention;

20       Fig. 7B schematically represents the distribution of route data from a route table manager Level-1 task secondary server to a route table manager Level-2 task client, according to an embodiment of the invention; and

      Fig. 8 schematically represents a series of steps involved in a method for synchronized distribution of routing data within a distributed processor  
25       router, according to another embodiment of the invention.

## **ACRONYMS**

The following acronyms and abbreviations are used in the description which follows:

BGP: Border gateway Protocol

5 CCB: Client Control Block

IS-IS: Intermediate System to Intermediate System

ITC: Inter task communication

LS: Location Service

OSPF: Open Shortest Path First

10 RTM: route table manager (also referred to as the global route table manager (GRTM)).

## **DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS**

15

In order to place the invention in perspective for the better understanding thereof, there now follows, with reference to Figs. 1A-1C, a brief description of a scalable router which may be used in conjunction with the instant invention.

Fig. 1A is a block diagram showing the basic architecture of a router 10. Each  
20 router 10 may include a plurality of shelves, represented in Fig. 1A as 20A to 20N. As shown in Fig. 1B, each shelf 20 can include a plurality of line cards, represented as 40A to 40N. For the purpose of clarity, only two control cards are shown in Fig. 1B; however, it is to be understood that in practice larger numbers of control cards can be used according to the invention. Each control  
25 card 30 is in communication with at least one line card 40. For example, control card 30A is shown as being in communication with line cards 40A and 40N on shelf 20A. Again, for the purpose of clarity, only two line cards are shown as being in communication with control card 30A. However, according to the invention, larger numbers of line cards may be connected to each control card.

Fig. 1C shows line card 40, which could be any of the line cards from a shelf of router 10. Line card 40 has a plurality of ports, or exterior interfaces, 50A, 50B, through 50N connected thereto. Although, only three  
5 interfaces are depicted in Fig. 1C, it is to be understood that a much larger number of interfaces may be used in practice.

### Introduction to a Route Table Manager

- 10 A route table manager (RTM) of the instant invention is a multifaceted software suite having a plurality of functions (tasks) that include, but are not necessarily limited to, the following:
1. messaging between RTM task servers and RTM task clients to form scalable and fault tolerant distribution topologies;
  - 15 2. managing exchange of database information between RTM tasks running on separate processors within a distributed processor environment;
  3. constructing a routing database from the sum of database information a) generated locally by tasks running on a local processor, and b) generated by and received from tasks running on at least one remote  
20 processor;
  4. constructing a forwarding database from the routing database; and
  5. propagating the forwarding database from RTM tasks having a higher hierarchical level (Level-1 tasks) to RTM tasks having a lower hierarchical level (Level-2 and lower-level tasks).

25

In a distributed multi-processor router, such as is encountered according to certain aspects of the instant invention, the RTM distributes information on dynamic routes, static routes, and interface information, hereafter referred to as database information. In return, RTM relies on a

number of tasks (services) for database information updates. Such tasks include those of dynamic routing protocols, IP, and an interface manager task. Routing protocols provide the RTM with updates on dynamic routes. IP tasks provide the RTM with updates on static routes. The interface  
5 manager task manages the ports, or external interfaces, of the router system, and provides the RTM with interface information. Interface information relates to a specific interface from which to dispatch a particular packet. Interface information, in general, is well known in the art.

- 10           The sum of the database information provided by services is collectively referred to as the routing database. Route entries maintained in the routing database include best and non-best routes. For example, all route entries that were injected by different routing protocols of the system's complement of routing protocols are stored in the routing database.
- 15   However, for a plurality of entries having the same destination prefix, only one of the entries is deemed the best. The decision as to which of those is the best entry (i.e. the best route for forwarding a packet) is based on a pre-configured preference value assigned to each routing protocol. For example, if static routes have a high preference value and IS-IS routes have a low  
20 preference value, and a route entry having the same destination prefix was injected by each protocol, although both entries will remain in the routing database, the static route is considered to be the best route. In embodiments of the invention, both the best routes and the non-best routes, as well as interface information, are retained in the routing database. A subset of the  
25 routing database exists which is referred to as the forwarding table. The forwarding table contains all route entries that are deemed the best plus all interface information. Therefore, according to the invention, both the best routes and the interface information define the forwarding table.

095606377 "06E300

A task of the RTM software suite typically runs on each of the plurality of processors of a multi-processor scalable system, including processors on control cards and line cards. The RTM task executing on each  
5 processor can be classified as either a Level-1 RTM task (L1) or a Level-2 RTM task (L2), and the processor may be termed an L1 or an L2 as a result. The distinction between an L1 and an L2 is in general the presence of either a routing database or a forwarding table. An L1 RTM task maintains the routing database and an L2 RTM task maintains the forwarding table. A  
10 subset of the plurality of processors of the system is statically configured to host an L1 RTM task and is referred to as the L1 pool. All other processors of the system outside of the L1 pool host an L2 RTM task.

As previously described, the RTM depends on a number of services for updates in routing information. A processor within the L1 pool may be  
15 running a number of such services, or none at all. Examples of such services include the IP routing protocols, OSPF, BGP, integrated ISIS, etc. (See, for example, C. Huitema, *Routing in the Internet*, 2<sup>nd</sup> Edition, Prentice Hall PTR, 2000.) According to the invention, each L1 is responsible for constructing a routing database from information generated in part by the  
20 local service(s), and in part from information generated by services running in association with other L1s. To obtain information that is generated by non-local services, i.e. information generated by services running on other L1s, an L1 must register with at least one other L1 where the service is running. According to the invention, in order to efficiently exchange  
25 locally generated information between L1s, each L1 can register with at least one other L1 as needed, on a per-service basis, to receive updates on the full complement of route data which is generated non-locally.

L1s register with each other for distribution of the following types of database information: dynamic routes including best and non-best routes, static routes including best and non-best routes, and interface information. An L1 is classified as an L1 server or L1 client for a given type of database information, depending on the existence of local services. An L1 task is an L1 server for a particular type of database information if the service which generates that information is running locally. An L1 task is an L1 client for a particular type of database information if the service which generates that information is not running locally and the L1 task has registered with an L1 server for information of this type. For example, if a BGP task was running on a given processor, the L1 task on that processor is considered an L1 server for BGP route information. If the same L1 task has registered with a remote L1 task for OSPF route information, the former L1 task is considered an L1 client of the remote L1 task with regard to OSPF route information.

Fig. 2 schematically represents exchange of route data, generated by different routing protocols, between a plurality of control cards 30A, 30B, and 30N within a distributed processor, scalable router, according to one embodiment of the invention. As alluded to hereinabove, the inventors have determined that superior performance from a scalable router is attained when routing protocols are distributed among control cards of the router. That is, superior performance is attained by running a plurality of different routing protocols on a plurality of processors within the control plane (on control cards) within the router. According to one embodiment, each of the plurality of processors is situated on a different control card of the router. With reference to Fig. 2, the plurality of control cards is represented by control cards 30A, 30B, and 30N. In the example shown in Fig. 2 a service or routing protocol task runs

on each control card 30A, 30B, 30N. Therefore, according to the definitions presented hereinabove, a Level-1 task (L1) of the RTM is running on each processor. In particular, according to the example shown in Fig. 2, control cards 30A, 30B, 30N run routing protocol A, routing protocol B, and routing protocol N, respectively. Routing protocol A, routing protocol B, and routing protocol N, provide route data A, route data B, and route data N, respectively. As described hereinabove, the L1 for each control card requires route data from the full complement of routing protocols running on the plurality of control cards 30A, 30B, and 30N. L1s therefore exchange route data by registering with other L1s on a per-service basis.

Fig. 3 schematically represents exchange of route data generated by two different routing protocols showing four servers and two clients, according to an embodiment of the invention. This aspect of the instant invention relates to the registration of L1s with at least one other L1, on a per-service basis, for the facile exchange of non-locally generated route data. Each entity I-IV represents an L1 task: L1A, L1A', L1B, and L1B', respectively. For the purpose of this example, the routing protocol tasks are designated as routing protocol A (RPA) in the case of L1A and L1A', and routing protocol B (RPB) in the case of L1B and L1B'. Under the control of the RTM, L1A registers as a client with both L1B and L1B' for information generated by routing protocol B, wherein both L1B and L1B' are servers. Similarly, L1B' registers as a client with both L1A and L1A' for information generated by routing protocol A, wherein both L1A and L1A' are servers. Thus, the same entity may have both client and server functionality concurrently. For the sake of clarity, L1A' and L1B are not shown as clients, but as servers only, therefore sending, rather than receiving information.

In the arrangement shown in Fig. 3 L1A is registered with both L1B and L1B', which both run RPB, and L1B' is registered with both L1A and L1A',

which both run RPA. This redundancy in preferred embodiments of the invention provides fault tolerance against the probability of failure of one or more L1 servers. Fault tolerance in the system is further described in a section below entitled Fault Tolerance.

5

Fig. 4 schematically represents the chronology of RTM-mediated data flow between control cards 30A and 30B of router 10, according to one embodiment of the invention. Only two control cards are depicted in Fig. 4, however it is to be understood that the principles of data flow could also apply to a larger number of control cards. Control cards 30A and 30B run services A and B, respectively. Each control card 30A and 30B also has an RTM task running, RTM A, RTM B, respectively. The fact of each of the processors running a service task dictates that RTM A and RTM B are both Level-1 as defined hereinabove. Data flow is initiated by information injection from service A to RTM A, as indicated by arrow 1. From RTM A, information is distributed concurrently to both route table A and to RTM B, as indicated by the two arrows each labeled 2. Thereafter, information is distributed from RTM B to route table B, as indicated by arrow 3. Finally, information is received by Service B from route table B, arrow 4. Data flow of the type illustrated in Fig. 4 enables the timely distribution of routing database updates between a plurality of control cards within a distributed processor router, in which the plurality of control cards are jointly responsible for running a plurality of different services.

By registration among L1s in the manner described herein, information generated by the full complement of services of the system can be effectively exchanged between L1s, with the result that each L1 maintains a synchronized routing database. Scalability of the distribution of database

information among L1s is achieved by the formation of distribution trees during the registration process.

According to the invention, each L1 task will maintain a  
5 synchronized copy of the routing database. Each L1 task has the role of constructing a synchronized forwarding table for use by L2 tasks, wherein the forwarding table consists of a subset of the routing database. Specifically, the routing database consists of all route entries, both best and non-best as defined above, as well as interface information. Each L1 is able  
10 to construct the forwarding table, based on the routing database, by identifying the best route for each destination prefix.

In this manner, when a best route is deleted from the routing database, each L1 can immediately replace the deleted "best route" with the  
15 next best route in the forwarding table which matches the particular destination prefix.

An L2 task is an RTM task which is running on a processor outside of the L1 pool. Each L2 requires a copy of the forwarding table. The  
20 source for forwarding table information are L1 tasks that are running throughout the system.

The hierarchical relationship of RTM tasks, according to a preferred embodiment of the invention, is schematically represented in Fig. 5. L1s  
25 represent the highest level, or top layer, of the hierarchical relationship. As described above, L1s are Level-1 RTM tasks which maintain a synchronized copy of the routing database and are the source of the forwarding table, whereas L2s are Level-2 RTM tasks which only maintain a copy of the

forwarding table. L2s themselves can occupy different hierarchical levels. In order to distinguish between L2s which occupy different hierarchical levels, L2 nodes which are clients of L1 servers as well as servers of L2 clients may be designated L2's; while L2s which are clients of L2' nodes may be designated L2"s. Thus, immediately below the L1s, at the intermediate hierarchical level or layer, lie L2s that are registered with L1s for forwarding table information. Below the intermediate hierarchical level lie L2's which are registered with an L2 node. Further, L2"s may be registered with L2's. According to a preferred embodiment, the depth of the topology shown in Fig. 5 is kept low by having a large fan-out at Layer 1. Again with reference to Fig. 5, it should be noted that although only a single server is shown for each client, according to a currently preferred embodiment of the invention designed for fault tolerance, i.e. tolerance of the router system to failure of a RTM task server, each client has at least two servers. In practice, for a given L2" client (Layer 4), one server can be a Layer 1 server (L1), and the other can be a Layer 2 node.

According to the invention, communication between RTM task clients and RTM task servers takes place to form scalable and fault tolerant distribution topologies. Among L1 tasks, distribution trees are formed for the propagation of routing database information. An L1 task which is running in association with a given service has the role of sourcing routing database information generated by that service. Distinct distribution trees therefore exist per service for the exchange of routing database information among L1 tasks. In a similar manner, distribution trees for the propagation of the forwarding table are formed with L1 tasks as the source of forwarding table information and L2 tasks as the nodes and leaves.

The RTM interacts with a Location Service module to determine the location of all RTM tasks running within router system 10. That is, the Location Service (LS) functions as a directory service. Interactions of the RTM with the LS include: (1) L1 RTM tasks, running on a control card 30, query the LS to determine the location of any RTM tasks acting as the source of routing database information for a particular service; (2) L2 RTM tasks query the LS to determine the location of any L1 RTM tasks (sources of forwarding table information); (3) LS notifies the RTM in the event that an RTM task comes up or goes down and (4) RTM tasks provide LS with RTM task type (including the routing database source) and level information to answer queries described in (1) through (3).

As described above, L1s are responsible for propagating the forwarding database to the Level-2 tasks (L2s). This is accomplished by the establishment of L1-L2 client-server relationships. L2 nodes register with L1s for the forwarding table only (i.e., L2 nodes register for the forwarding table "service"). According to one aspect of the invention, an L1 server will accept N L2 clients, where N is determined, at least in part, by the configured maximum fan-out. This situation is schematically represented in Fig. 6, in which an L1 server (L1A) already has N L2 clients, represented by L2A, L2B, and up to L2N. Client M represents an L2 that is not a client of an RTM task running in the control plane of the router system. If client M then signals a request to register with L1A (arrow 1), that request is denied as represented by arrow 2. If maximum fan-out has been reached on all L1s in the control plane, client M then requests registration (arrow 3) with an L2, e.g. L2A, that is a client of an L1 (in this case L1A). A registration response message is then sent from L2A to client M, as represented by arrow 4. Client M can now receive forwarding table updates from L1A via L2A.

Maximum fan-out in L1-L2 client-server relationships is determined, *inter alia*, by CPU load. In case maximum fan-out of all L2 servers has been reached, then a client can force registration. This client-server registration procedure is used to form distribution trees for the propagation of the

5 forwarding database among all L2 clients. Information on the location of the servers is available from the LS. According to a currently preferred embodiment, the LS itself runs on all control cards 30 and line cards 40 of router system 10.

10 It will be apparent to the skilled artisan that the client-server registration procedure described here is hierarchically based, in that L2s first attempt to register with L1s until maximum fan-out has been reached, and only then will an L2 attempt to register with an L2 that is registered as a client of an L1. An L2 which acts as a server to an L2 client may be  
15 designated L2', and an L2 client of an L2' server may be designated L2" (Fig. 5). Large scale distribution is therefore achieved by using a reliable multicast transmission at the tree nodes. In general, the number of L2s is greater than the number of L1s. According to one embodiment, the ratio of L1s to L2s ranges from about 1:1 to about 1:15.

20

### **Fault Tolerance**

Fault tolerance in the system of the invention, as alluded to briefly above, is achieved by redundancy in registration, and therefore in  
25 communication. As a client, an L1 or L2 task registers with at least two servers from which it may receive the same information. One of the servers with which the client registers is considered a primary server, and the other a secondary. The client communicates exclusively with the primary unless

and until the primary fails in some manner to deliver, and then the client turns to the secondary for database updates. Service is thus uninterrupted.

In the event of a server failure, and a necessary switchover by a client to its secondary server, the client receives a copy of the secondary server's database. If the client is a node in a distribution tree, it simply delivers to its clients the difference between the existing database and the copy of the database received from the secondary server.

Referring now to Fig. 7A, the role of a control card as a Level-2 node is to receive forwarding entries from its primary L1 server, and then to redistribute the forwarding entries to its own clients, represented as L2 clients A, B, and C. The L2 node is registered with two L1 servers, the primary L1 server and the secondary L1 server, for the purpose of fault tolerance, as schematically represented in Fig. 7A.

Referring now to Fig. 7B, if the primary L1 server fails, the L2 node activates its secondary L1 server. When the secondary L1 server is activated, it delivers a complete copy of its database to the L2 node, as schematically represented in Fig. 7B. When the L2 node receives the copy of the entire table from the secondary L1 server, it compares that copy to its existing database, and calculates the difference between the two. It only needs to distribute to L2 clients A, B and C the difference between the entire new table and its existing table.

25

Fig. 8 schematically represents a series of steps in a method for the synchronized distribution of routing data within a distributed processor, highly-scalable router, according to one embodiment of the invention. Step 800 of Fig. 8 involves running at least one routing protocol of a complement

of routing protocols on individual ones of a first plurality of processors, wherein each routing protocol of the complement of routing protocols generates routing data. This first plurality of processors are the L1 processors described in detail above. Also as previously described, it is the configuration of the L1s to run routing protocols and to otherwise behave as L1s that makes them L1s. An L1 may not be running a routing protocol, but still be an L1. That is, an L1 may obtain all of its routing data from other L1s with which it registers as a client.

Step 802 involves registering each of the first plurality of processors with at least one other of the first plurality of processors. Step 804 involves exchanging the routing data between members of the first plurality of processors, such that each of the first plurality of processors receives a full complement of routing data generated by the complement of routing protocols. The complement of routing data received by each of the first plurality of processors provides a complete routing database. Step 806 involves forming a forwarding database from the complete routing database provided as a result of step 804. The forwarding database formed in step 806 is comprised of a subset of the complete routing database provided in step 804.

Step 808 involves propagating the forwarding database from the first plurality of processors to a second plurality of processors of the distributed processor router, wherein the second plurality of processors are characterized as not running (or being configured to run) routing protocols. The method steps 800 through 808 may be sequentially repeated over time, for example, when updated reachability information is received from one or more peer routers of the distributed processor router.

# Introduction

5

10